REMARKS

This paper is being provided in response to the May 3, 2006 Office Action for the above-referenced application. In this response, applicant has made minor modifications to the Abstract and has amended claims 1, 8 and 10 in order to clarify that which applicant deems to be the invention. Applicant respectfully submits that the changes to the specification do not add new matter and that the modifications to the claims are all supported by the originally-filed application. See, for example, the discussion on page 12 beginning at line 9 of the originally-filed specification.

The objection to the specification has been addressed by amendments provided herein in accordance with the guidelines set forth in the Office Action. Accordingly, applicants respectfully request that this objection be withdrawn.

The rejection of claims 1-18 under 35 U.S.C. 101 as being directed to non-statutory subject matter has been addressed by amendments provided herein in accordance with the guidelines set forth in the Office Action. In this regard, applicant notes that claims 1-9, as presently configured, recite a physical transformation of data provided to storage devices and that MPEP 2106 IV.B.2.(b) provides that a claimed process is clearly statutory if it results in a physical transformation outside the computer. Claims 10-18, as presently configured, are directed to a computer readable storage medium. Accordingly, applicants respectfully request that this rejection be withdrawn.

The rejection of claims 1-7 and 10-16 under 35 U.S.C. 103(a) as being unpatentable over the article "File System Design for an NFS Server" by Hitz, et al. (hereinafter "File System Design article") in view of U.S. Patent No. 5,819,292 to Hitz et al. (hereinafter "Hitz") is hereby traversed and reconsideration thereof is respectfully requested.

Claim 1, as amended herein, recites a method of restoring data to a first storage device. The method includes providing data in the first storage device at a first storage area of a first type that contains sections of data, providing data in a second storage device at a second storage area of a second type where the second type has, for each section of data thereof, at least one of: a pointer to a corresponding section of data of the first storage area and a pointer to corresponding section of data of a third storage device at a third storage area of the first type, wherein prior to writing new data to a section of the first storage area pointed to by a pointer of the second storage area, data of the section of the first storage area is copied to a section of the third storage area and the pointer of the second storage area is adjusted to point to the section of the third storage area, providing data in a fourth storage device having at least one other storage area of the second type, and, for each particular section of data of the second storage area having a pointer to the third storage area, providing to a corresponding section of the first storage area an indirect pointer to a corresponding section of the third storage area if no storage areas of the at least one other storage area point to the corresponding section of the first storage area. Claims 2-7 depend from claim 1.

Claim 10, as amended herein, recites computer software, provided in a computer-readable storage medium, that restores data to a first storage area of a first type that contains sections of

data from a second storage area of a second type that has, for each section of data thereof, at least one of: a pointer to a corresponding section of data of the first storage area and a pointer to corresponding section of data of a third storage area of the first type where there is at least one other storage area of the second type. The software includes executable code that, prior to writing new data to a section of the first storage area pointed to by a pointer of the second storage area, copies data of the section of the first storage area to a section of the third storage area and adjusts the pointer of the second storage area to point to the section of the third storage area, executable code that iterates through each section of the second storage area, and executable code that provides to a corresponding section of the first storage area an indirect pointer to a corresponding section of the third storage area if no storage areas of the at least one other storage area point to the corresponding section of the first storage area. Claims 11-16 depend from claim 10.

The File System Design article discloses a method for restoring data using a snapshot technique. Section 3.4 beginning on page 10 illustrates the snapshot technique where figure 3(a) shows the file system prior to creating the snapshot, figure 3(b) shows the file system after creating the snapshot, and figure 3(c) shows what happens when a user modifies a data block.

Hitz discloses a method for maintaining consistent states of a file system. Figures 18A-18C illustrate a snapshot operation where a snapshot inode 1822 is provided as a snapshot of the file system block 1810. Note that figures 18A-18C of Hitz are substantially similar to figures 3(a)-3(c) of the File System Design article. Similarly, the corresponding descriptions of how snapshots are created and handled in both Hitz and the File System Design article are

substantially similar. Accordingly, the detailed description below of Hitz is also applicable to the File System Design article.

As described in column 19 of Hitz, figure 18C is a diagram illustrating the snapshot 1822 when a change to the active file system occurs after the snapshot 1822 is taken (created). Figure 18C show an instance where a block 1818 comprising data "D" is modified after the snapshot was taken (in figure 18B). A new block 1824 containing new data D' is allocated and pointed to by the file block 1810 while the snapshot block 1822 maintains its pointer to the block 1818 containing the old data, D. After the snapshot and subsequent write to the block 1818 containing the data, D, the active file system comprises blocks 1812, 1814, 1816, 1820, 1824 but does not contain the block 1818 containing the data, D. However, the block 1818 containing the data D is not overwritten. Rather, the block 1818 is protected against being overwritten by a snapshot bit that is set for the entry for the block 1818. Thus, the snapshot 1822 points to the unmodified block 1818 as well as the other blocks 1812, 1814, 1816, 1820. In effect, a write to the block 1818 containing the data, D, causes the data D to be copied to a new block 1824 that is pointed to by the file system block 1810 while the snapshot 1822 maintains its previous pointer to the old block 1818.

Neither the File System Design article nor Hitz, alone or in any combination, show, teach or suggest the feature recited in Applicants' independent claims where prior to writing new data to a section of the first storage area pointed to by a pointer of the second storage area, data of the section of the first storage area is copied to a section of the third storage area and the pointer of the second storage area is adjusted to point to the section of the third storage area. This recited

feature may be understood by reviewing Applicant's specification at, for example, figure 3 which shows the virtual storage area (84) pointing to both the standard logical device (82) and the log device (86). Each of the pointers from the virtual storage device (84) to the log device (86) correspond to a write to a corresponding section of the standard logical device (82). Thus, in response to a write to a section of the standard logical device (82), the old data is first copied to the log device (86), the corresponding pointer is made to point to the log device (86), and then the write is provided to the standard logical device (82).

In contrast to the feature recited in applicant's independent claims, discussed above, both the File System Design article and Hitz discloses that, when data is written to the file system, the data is copied to a new location (e.g., copied from the old block 1818 to the new block 1824 in figure 18C) and the device to which the write occurred is made to point to the new data block 1824. Thus, unlike applicant's claimed invention where the second storage area points to the moved old data, Hitz teaches the opposite where the original device to which the write is being made points to a different block that is allocated. The disadvantage of the system disclosed in Hitz is that the existence of the snapshot inode 1822 causes a change in the operation of the standard device 1810 (e.g., pointing to the block 1824 instead of the block 1818 when a write occurs). That is, if the snapshot inode 1822 of figure 18C were not present in Hitz, the block 1810 would point to the block 1818 rather than pointing to a new block 1824. In contrast, the present claimed invention does not cause any alteration in the location of the data of the standard device because all of the changes due to writes are handled by the virtual device and log device.

In other words, independent claims 1 and 10 recite, in some form, that a write to the first storage area causes the old data to be copied to a new location and a pointer of the second storage area to point to the old data. In contrast, both the File System Design article and Hitz specifically teach that, in response to a write to what is, in effect, the first storage area (illustrated by the block 1810), the data is copied from the old block 1818 to a new block 1824 and the first storage area is made to point to the new block 1824. This is different than applicant's claimed invention and is less than optimal because the existence of the snapshot inode 1822 causes alterations in the structure and operation of the first storage area (1810).

Based on the above, Applicants respectfully request that this rejection be reconsidered and withdrawn.

The rejection of claims 8, 9, 16 and 17 under 35 U.S.C. 103(a) as being unpatentable over the File System Design article in view of Hitz and further in view of U.S. Patent No. 6,460,054 to Grummon (hereinafter "Grummon") is hereby traversed and reconsideration thereof is respectfully requested.

Claims 8 and 9 depend from claim 1, discussed above. Claims 16 and 17 depend from claim 26, discussed above. The File System Design article and Hitz are discussed above. The Office Action indicates that Grummon teaches a system for management of storage devices where storage areas and data sections are defined corresponding to tracks/storage devices.

Applicants respectfully submit that the deficiencies of the the File System Design article and Hitz with respect to claims 1 and 10, discussed above, are not overcome by the addition of the Grummon reference. Accordingly, Applicants respectfully request that this rejection be withdrawn.

Based on the above, Applicants respectfully request that the Examiner reconsider and withdraw all outstanding rejections and objections. Favorable consideration and allowance are earnestly solicited. Should there be any questions after reviewing this paper, the Examiner is invited to contact the undersigned at 508-898-8603.

Respectfully submitted,
MUIRHEAD AND SATURNELLI, LLC

Donald W. Muirhead
Registration No. 33,978

August 2, 2006
Date

Muirhead and Saturnelli, LLC
200 Friberg Parkway, Suite 1001
Westborough, MA 01581
508-898-8601 (main)
508-898-8602 (fax)
Customer No. 26339